

## Отчет о проведении тестирования.

Время проведения: 29.04.2021

### 1. Цель теста

Бизнес-цель теста – *получение навыков работы с индексами и нагрузочного тестирования.*

Техническая цель теста – оценка производительности с использованием *wrk* с использованием СУБД MySQL 8.0.23.

*В испытании участвовало 6 операций.*

*Таблица 1. Модель нагрузки.*

|          |  |
|----------|--|
| <i>1</i> | <i>Нагрузка настройка MySQL по умолчанию. Без использования индексов.</i>                |
| <i>2</i> | <i>Нагрузка настройка MySQL по умолчанию. С использованием индексов.</i>                 |
| <i>3</i> | <i>Нагрузка с размером innodb buffer pool size = 128 Mb. Без использования индексов.</i> |
| <i>4</i> | <i>Нагрузка с размером innodb buffer pool size = 128 Mb. С использованием индексов.</i>  |
| <i>5</i> | <i>Нагрузка с размером innodb buffer pool size = 348 Mb. Без использования индексов.</i> |
| <i>6</i> | <i>Нагрузка с размером innodb buffer pool size = 348 Mb. С использованием индексов.</i>  |

*Было произведено 4 эксперимента с разным количеством запросов в секунду.*

*Таблица 2. Эксперименты.*

|          |                                  |
|----------|----------------------------------|
| <i>1</i> | <i>1 поток 1 соединение</i>      |
| <i>2</i> | <i>2 потока 10 соединений</i>    |
| <i>3</i> | <i>4 потока 100 соединений</i>   |
| <i>4</i> | <i>8 потоков 1000 соединений</i> |

### 2. Выводы

1. Максимальная производительность системы – *57* запросов в секунду при 8-ми потоковом тесте и *100* соединениях.
2. Время *отклика / выполнения* на уровне максимальной производительности по операциям составляло от *691,21* миллисекунд до *5,3* секунды. Так же были просадки после *90 перцентиля* в диапазоне *11-24* секунды.
3. До применения индекса нагрузка в *1000* запросов в секунду практически не работала. Количество ответов на запросы с ошибками или таймаутами доходило до 100% с размером пула меньше *348 Mb*, При размере буффер-пула в *348 Mb* и пула-сортировки *8 Mb* количество ошибок в ответах удалось сократить до *7,8%*, после применения индекса.

## Настройки, запрос, создание индекса

4. При тестировании использовался docker image: mysql:8.0.23

5. Первые две операции проводились с настройками MySQL:

```
max_connections = 1088
innodb_flush_method = O_DIRECT
```

6. Последующие операции с настройками:

```
innodb buffer pool size 128M
max_connections = 1088
innodb_flush_method = O_DIRECT
innodb_buffer_pool_size = 134217728
innodb buffer pool size 348M
max_connections = 1088
innodb_flush_method = O_DIRECT
innodb_buffer_pool_size = 364904448
sort_buffer_size = 2097152
```

7. Создание индекса проводилось следующим запросом:

```
ALTER TABLE user ADD INDEX idx_user_surname_name_1342 USING
BTREE (surname, name) COMMENT 'with index selectivity';
Селективность индекса выбиралась из кардинальности по полей surname и name:
SELECT COUNT(*) c
FROM (SELECT surname, COUNT(*) FROM user GROUP BY surname) t;
c: 470
SELECT COUNT(*) c
FROM (SELECT name, COUNT(*) FROM user GROUP BY name) t;
c: 3006
```

8. Запрос и его план выполнения после создания индекса:

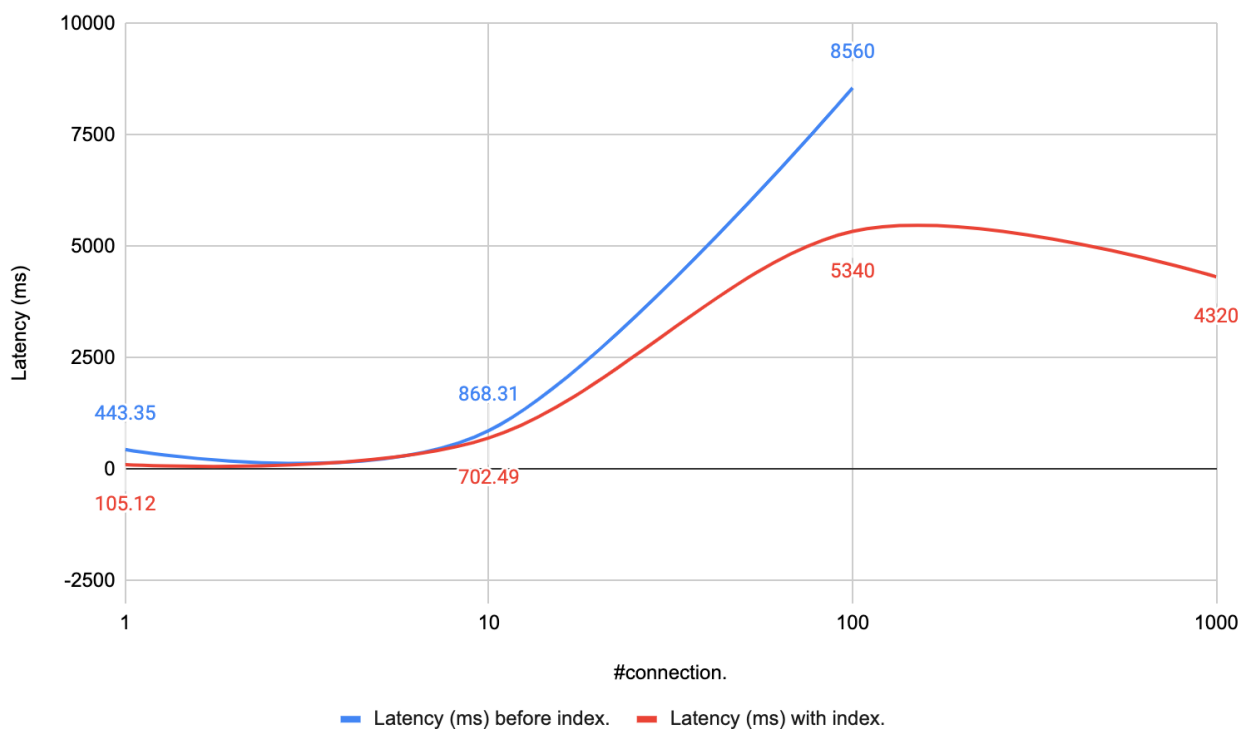
```
EXPLAIN FORMAT=TREE
SELECT u.id, username, name, surname, age, sex, city,
       JSON_ARRAYAGG(interests), NOT isnull(uhf.id) AS is_friend
FROM user u
LEFT JOIN user_has_interests uhi ON uhi.user_id = u.id
LEFT JOIN interest i ON i.id = uhi.interest_id
LEFT JOIN user_has_friends uhf ON uhf.friend_id = u.id
AND uhf.user_id = 0x...
WHERE name LIKE 'a%' AND surname LIKE 'b%'
GROUP BY u.id, username, name, surname, age, sex, city, uhf.id;;
```

```
| -> Group aggregate: json_arrayagg(interest.interests)
-> Sort: u.id, u.username, u.`name`, u.surname, u.age, u.sex, u.city, uhf.id
-> Stream results (cost=142760.58 rows=347056)
-> Nested loop left join (cost=142760.58 rows=347056)
-> Nested loop left join (cost=69497.05 rows=347056)
-> Left hash join (uhi.user_id = u.id) (cost=34789.45 rows=347056)
-> Index range scan on u using idx_user_surname_name_1342, with index condition: ((u.`name` like 'a%') and
(u.surname like 'b%')) (cost=87686.47 rows=173528)
-> Hash
-> Table scan on uhi (cost=0.00 rows=2)
-> Single-row index lookup on i using PRIMARY (id=uhi.interest_id) (cost=0.00 rows=1)
-> Filter: ((uhf.user_id = 0x...) and (uhf.friend_id = u.id)) (cost=1.00 rows=1)
-> Index lookup on uhf using fk_user_has_friends_0242 (user_id=...) (cost=1.00 rows=1)
```

## 9. Графики

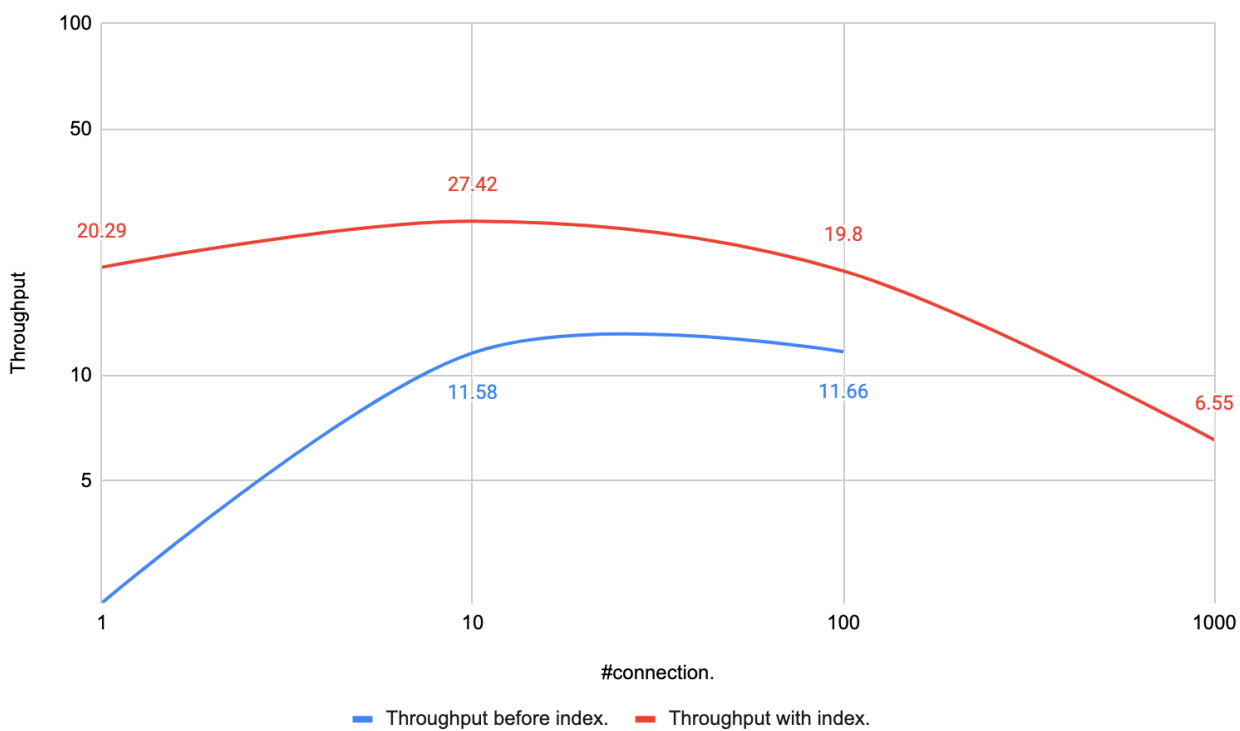
Настройки по умолчанию. По оси x- кол-во соединений, по оси y latency.

График 1. Время отклика по количеству соединений.



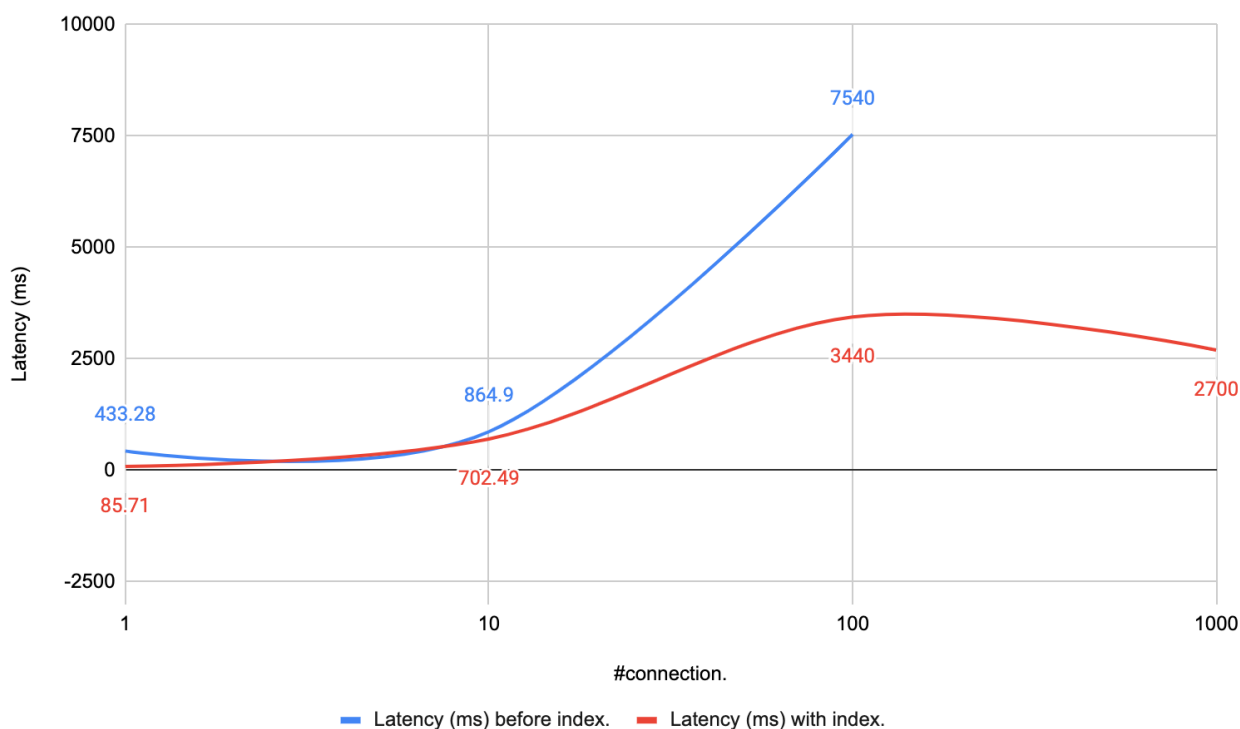
Настройки по умолчанию. По оси x- кол-во соединений, по оси y throughput.

График 2. Пропускная способность по количеству соединений.



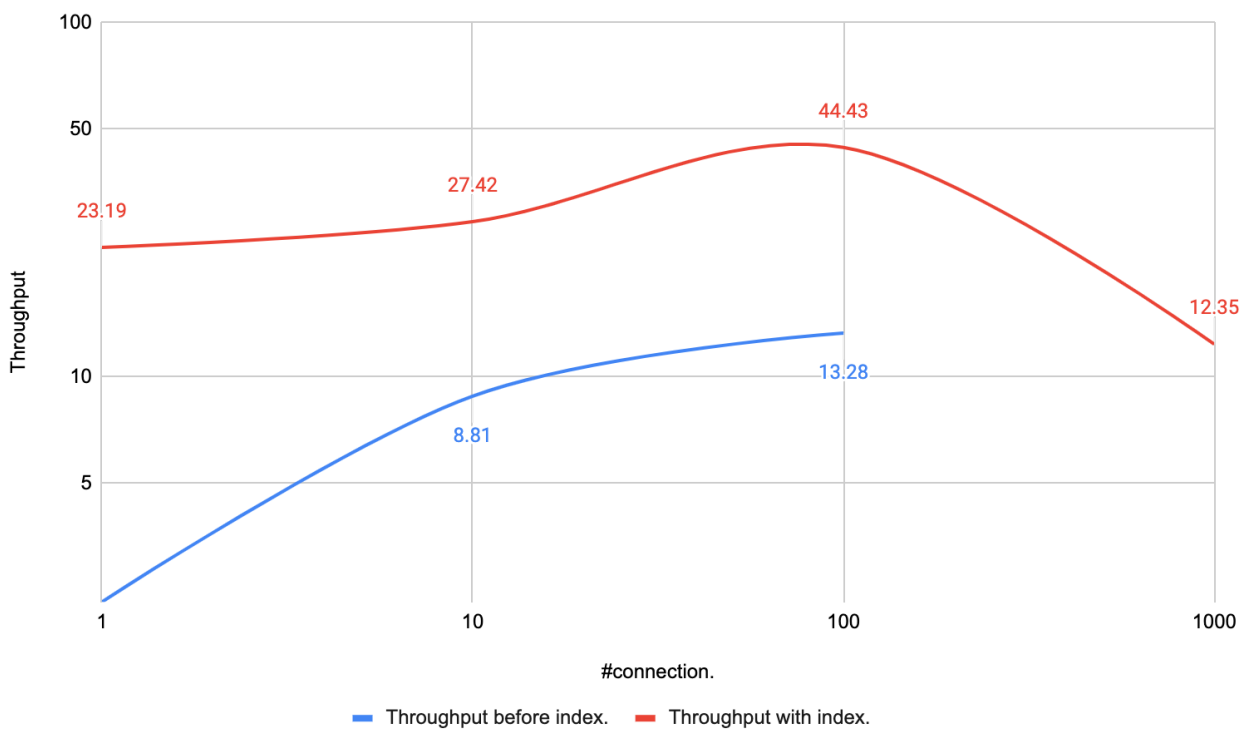
По оси x- кол-во соединений, по оси y latency (innodb buffer pool size 128 Mb).

График 1. Время отклика по количеству соединений.



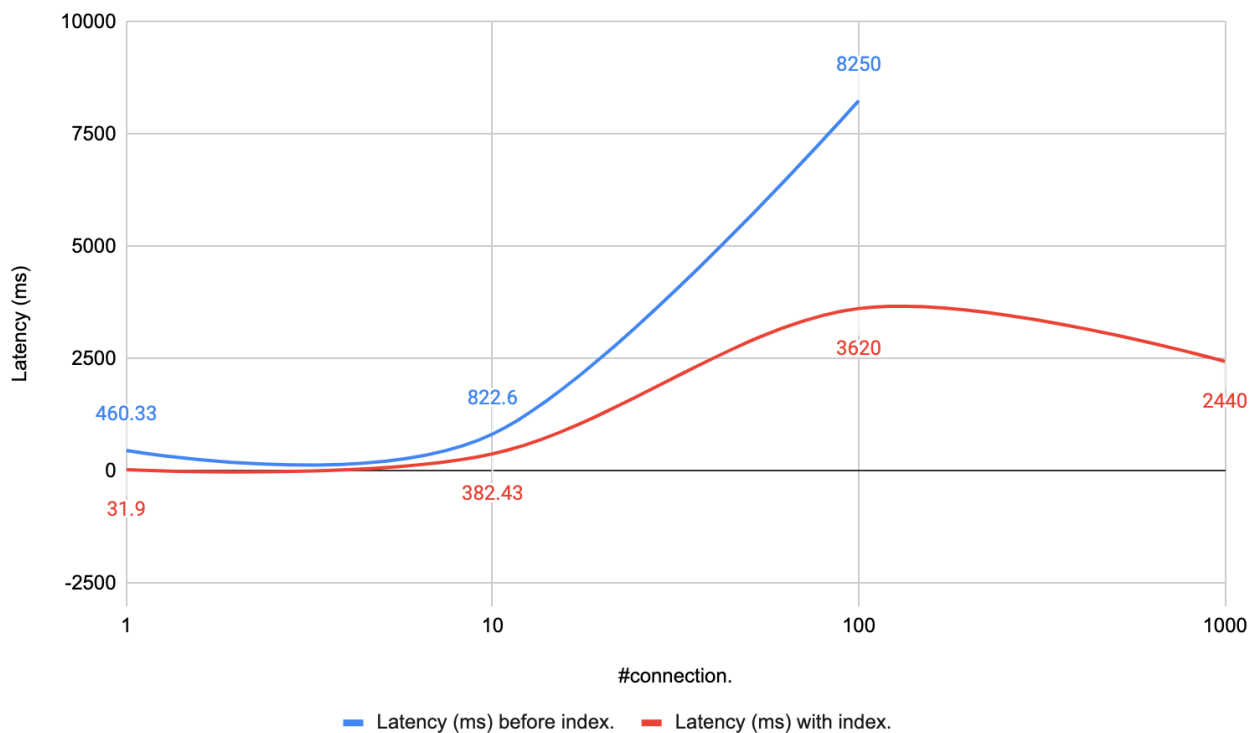
По оси x- кол-во соединений, по оси y throughput (innodb buffer pool size 128 Mb).

График 2. Пропускная способность по количеству соединений.



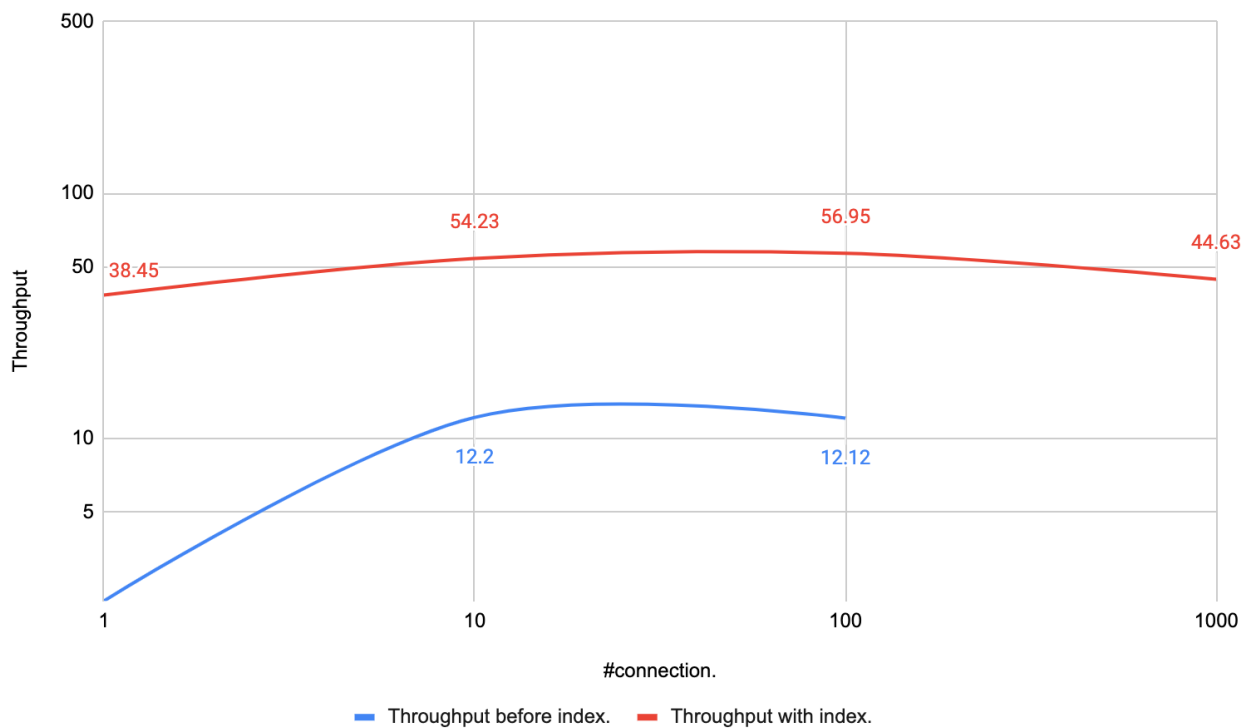
По оси x- кол-во соединений, по оси y latency (innodb buffer pool size 348 Mb).

График 1. Время отклика по количеству соединений.



По оси x- кол-во соединений, по оси y throughput (innodb buffer pool size 348 Mb).

График 2. Пропускная способность по количеству соединений.



## 10. Запрос и его план выполнения после создания индекса (формат JSON):

```
EXPLAIN FORMAT=JSON
```

```
SELECT u.id, username, name, surname, age, sex, city,  
       JSON_ARRAYAGG(interests), NOT isnull(uhf.id) AS is_friend  
FROM user u  
LEFT JOIN user_has_interests uhi ON uhi.user_id = u.id  
LEFT JOIN interest i ON i.id = uhi.interest_id  
LEFT JOIN user_has_friends uhf ON uhf.friend_id = u.id  
  AND uhf.user_id = 0x5CB9CAE1977911000000000000000000  
WHERE name LIKE 'a%%' AND surname LIKE 'b%%'  
GROUP BY u.id, username, name, surname, age, sex, city, uhf.id;;
```

```
{  
  "query_block": {  
    "select_id": 1,  
    "cost_info": {  
      "query_cost": "137816.82"  
    },  
    "grouping_operation": {  
      "using_temporary_table": true,  
      "using_filesort": true,  
      "nested_loop": {  
        {  
          "table": {  
            "table_name": "u",  
            "access_type": "range",  
            "possible_keys": [  
              "idx_user_surname_name_1342"  
            ],  
            "key": "idx_user_surname_name_1342",  
            "used_key_parts": [  
              "surname"  
            ],  
            "key_length": "3078",  
            "rows_examined_per_scan": 173528,  
            "rows_produced_per_join": 19278,  
            "filtered": "11.11",  
            "index_condition":  
              "((('hl`.`u`.`name` like 'a%%') and ('hl`.`u`.`surname` like 'b%%')))",  
            "cost_info": {  
              "read_cost": "85677.77",  
              "eval_cost": "1927.90",  
              "prefix_cost": "87605.67",  
              "data_read_per_join": "83M"  
            },  
            "used_columns": [  
              "id",  
              "username",  
              "name",  
              "surname",  
              "age",  
              "sex",  
              "city"  
            ],  
          },  
        },  
        {  
          "table": {  
            "table_name": "uhi",  
            "access_type": "ALL",  
            "possible_keys": [  
              "fk_user_has_interests_9774"  
            ],  
            "rows_examined_per_scan": 2,  
            "rows_produced_per_join": 38557,  
            "filtered": "100.00",  
            "using_join_buffer": "hash join",  
            "cost_info": {  
              "read_cost": "83.85",  
              "eval_cost": "3855.79",  
              "prefix_cost": "91545.31",  
              "data_read_per_join": "2M"  
            },  
            "used_columns": [  
              "id",  
              "user_id",  
              "interest_id"  
            ],  
            "attached_condition":  
              "<?if>(is_not_null_compl(uhi), ('hl`.`uhi`.`user_id` = 'hl`.`u`.`id`), true)"  
            },  
          },  
        },  
      },  
    },  
  },  
}
```

```
{  
  "table": {  
    "table_name": "i",  
    "access_type": "eq_ref",  
    "possible_keys": [  
      "PRIMARY"  
    ],  
    "key": "PRIMARY",  
    "used_key_parts": [  
      "id"  
    ],  
    "key_length": "16",  
    "ref": [  
      "hl.uhi.interest_id"  
    ],  
    "rows_examined_per_scan": 1,  
    "rows_produced_per_join": 38557,  
    "filtered": "100.00",  
    "cost_info": {  
      "read_cost": "2.00",  
      "eval_cost": "3855.79",  
      "prefix_cost": "95403.10",  
      "data_read_per_join": "20M"  
    },  
    "used_columns": [  
      "id",  
      "interests"  
    ],  
  },  
},  
{  
  "table": {  
    "table_name": "uhf",  
    "access_type": "ref",  
    "possible_keys": [  
      "fk_user_has_friends_0242",  
      "fk_user_has_friends_7122"  
    ],  
    "key": "fk_user_has_friends_0242",  
    "used_key_parts": [  
      "user_id"  
    ],  
    "key_length": "16",  
    "ref": [  
      "const"  
    ],  
    "rows_examined_per_scan": 1,  
    "rows_produced_per_join": 38557,  
    "filtered": "100.00",  
    "cost_info": {  
      "read_cost": "38557.92",  
      "eval_cost": "3855.79",  
      "prefix_cost": "137816.82",  
      "data_read_per_join": "2M"  
    },  
    "used_columns": [  
      "id",  
      "user_id",  
      "friend_id"  
    ],  
    "attached_condition":  
      "<?if>(is_not_null_compl(uhf), (('hl`.`uhf`.`user_id` = 0x5cb9cae1977911000000000000000000) and  
      ('hl`.`uhf`.`friend_id` = 'hl`.`u`.`id`)), true)"  
    },  
  },  
},  
}
```